

101

Fig. 1

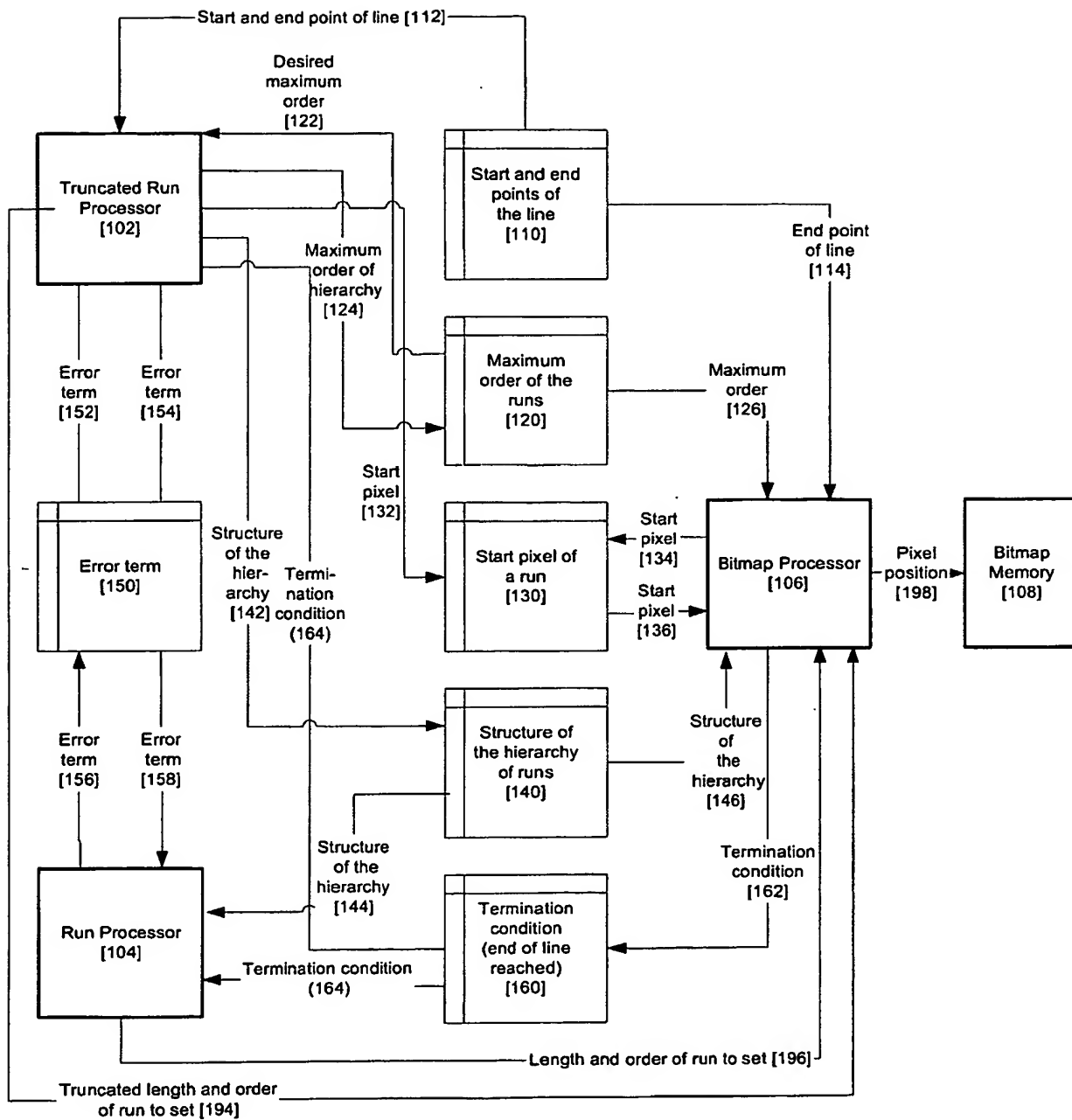
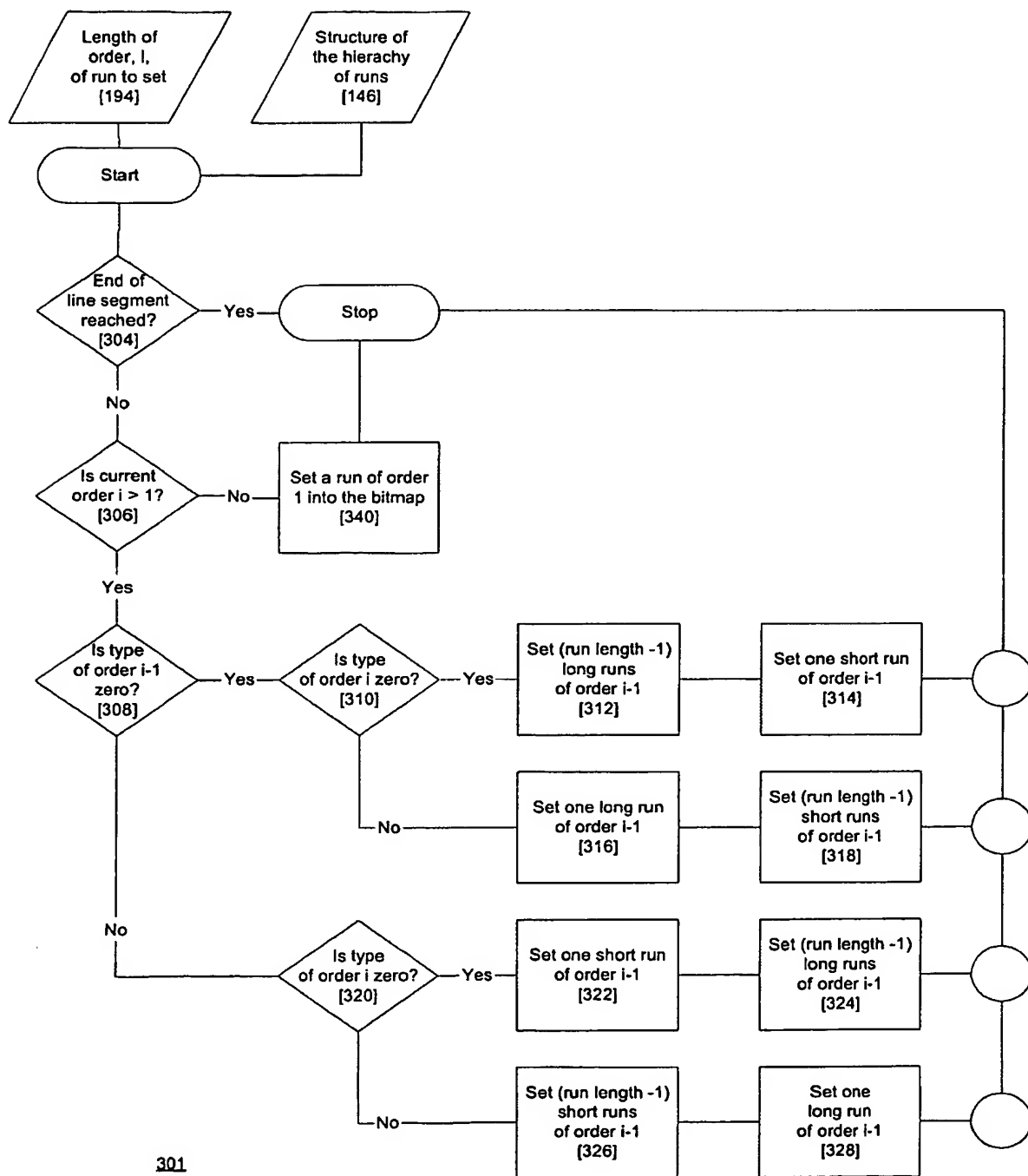
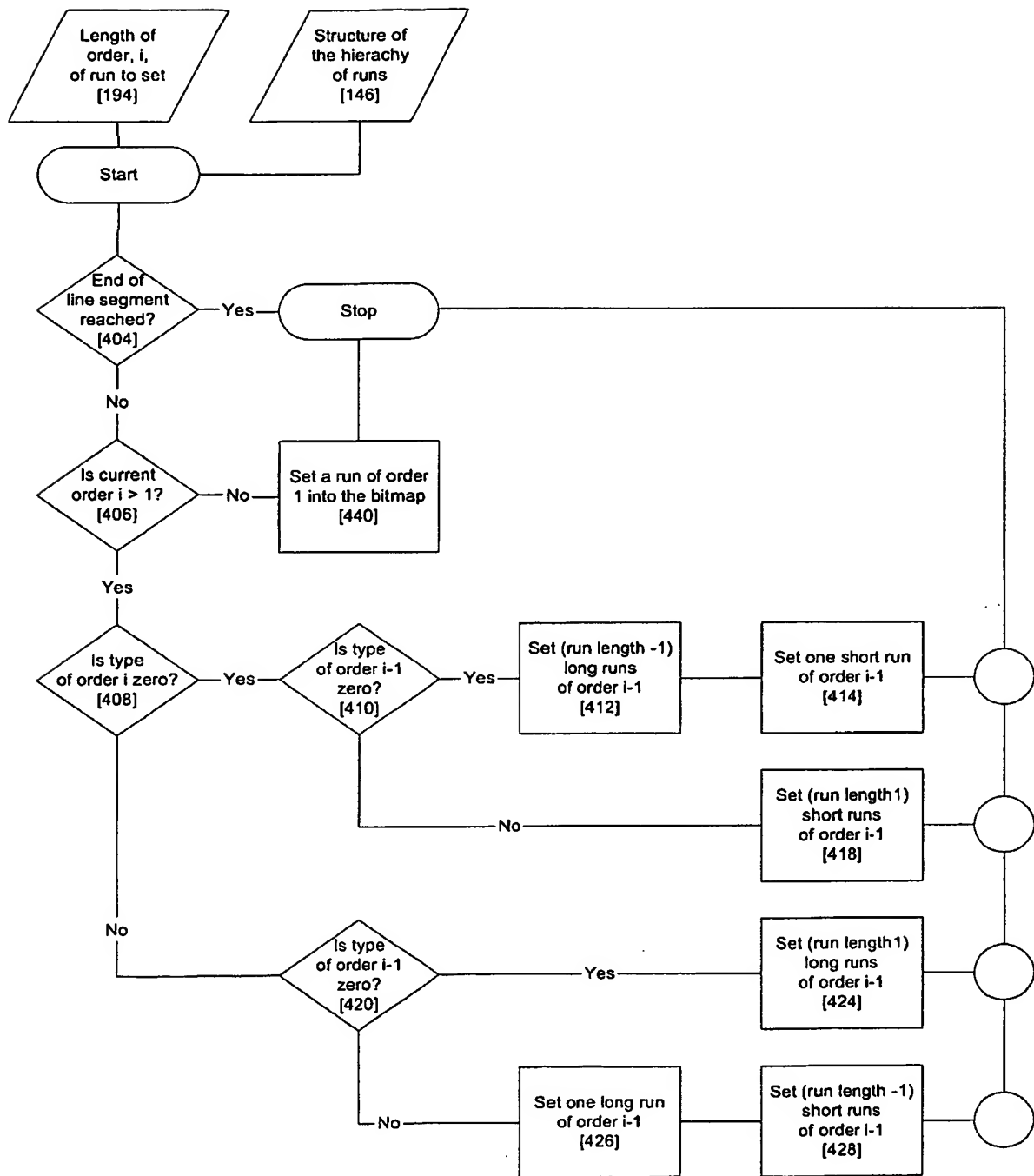


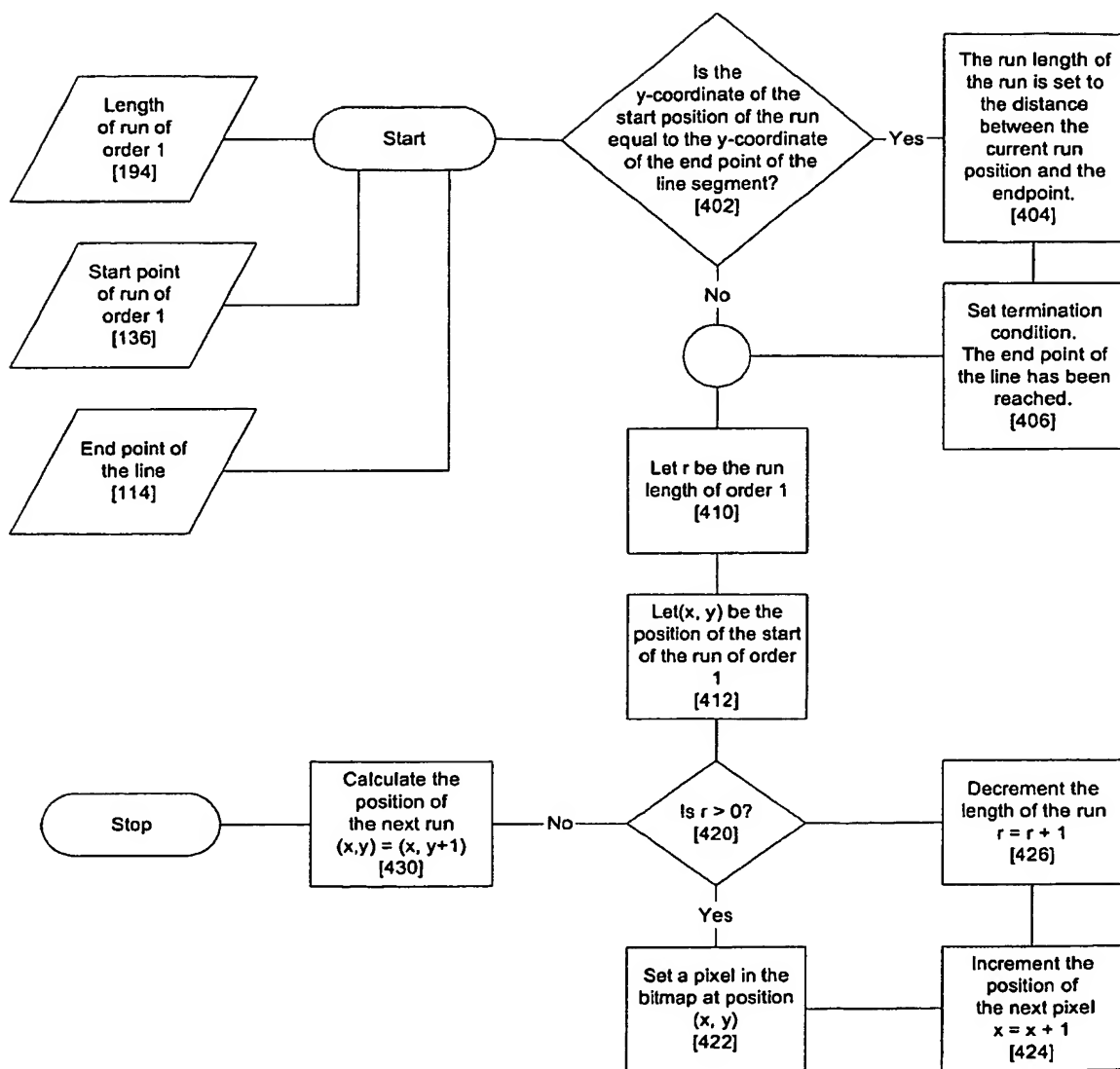
Fig. 2

Fig. 3



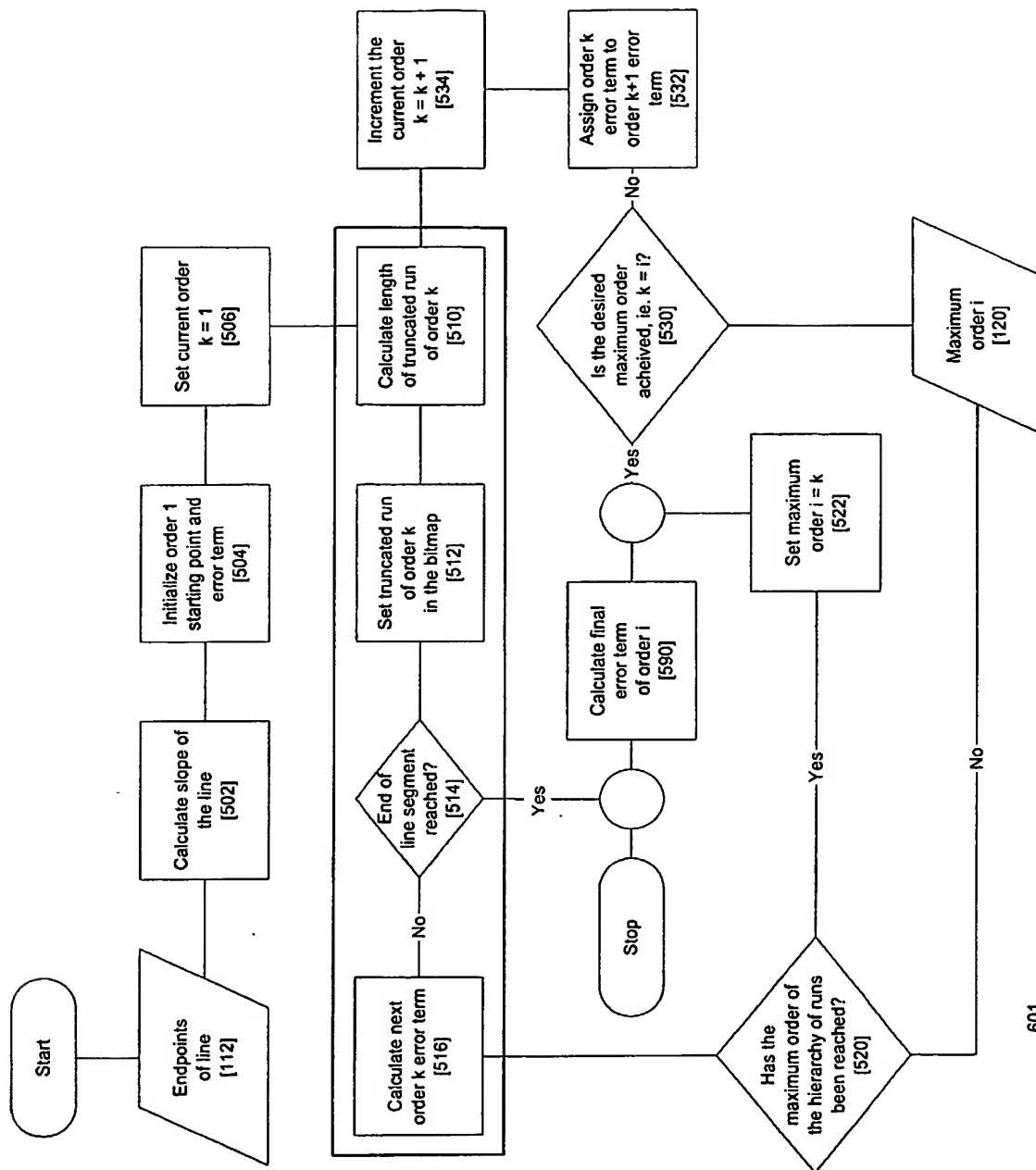
401

Fig. 4



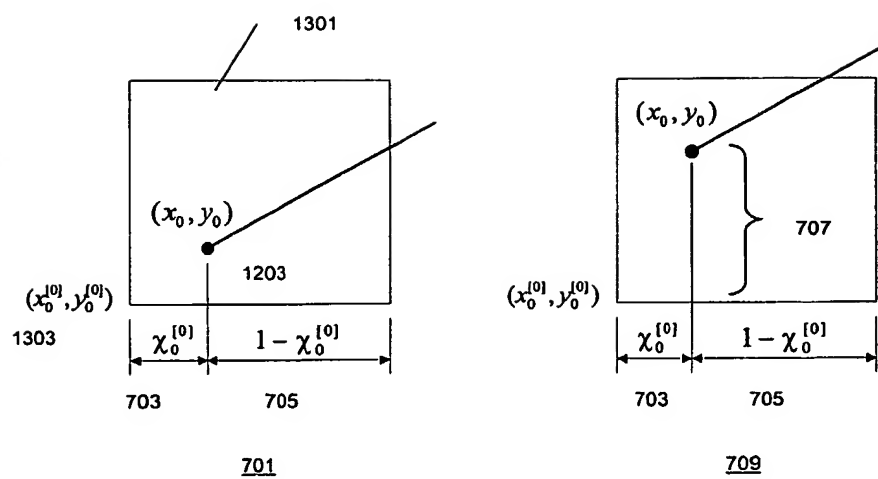
501

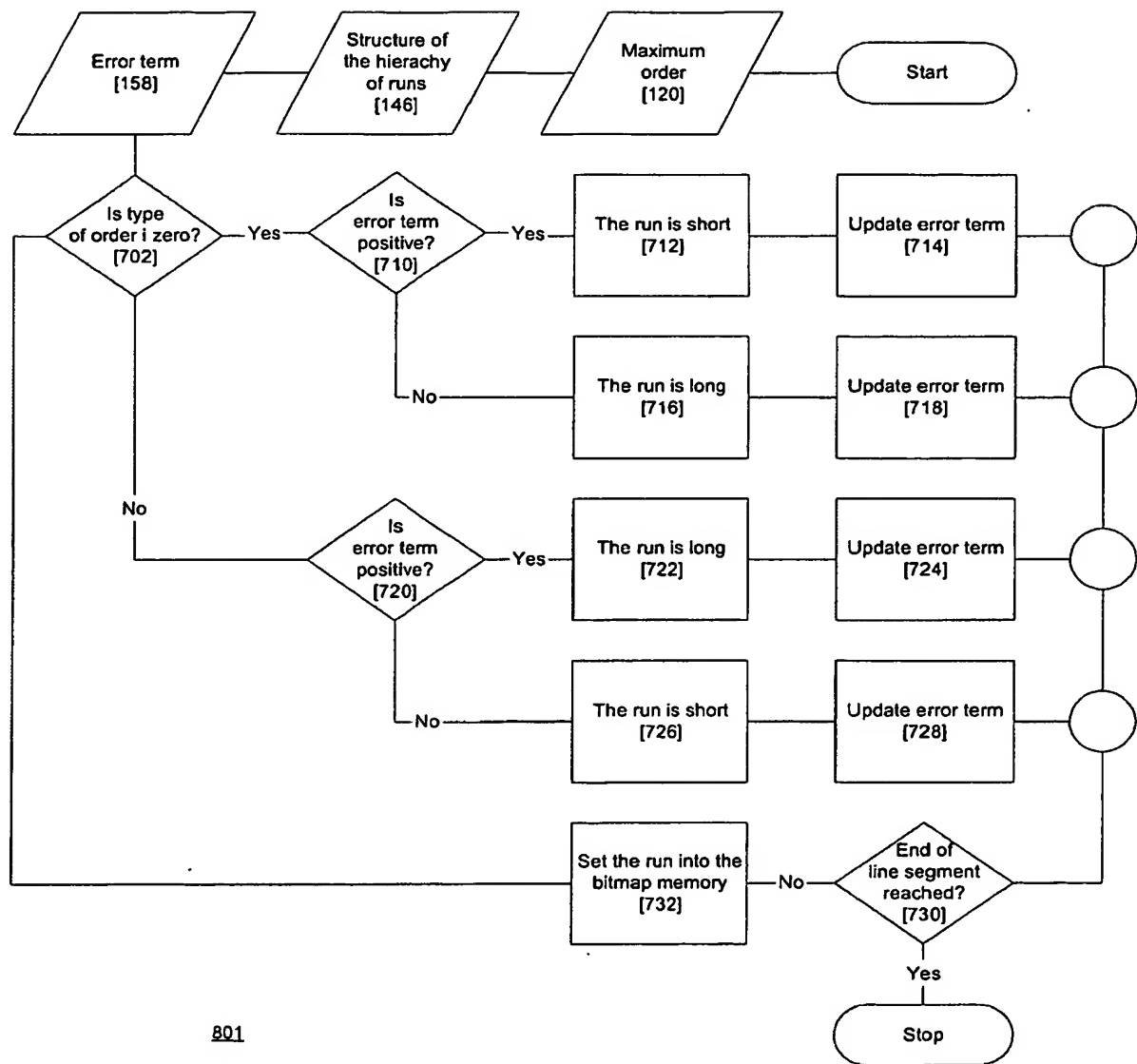
Fig. 5



601

Fig. 6

Fig. 7



801

Fig. 8

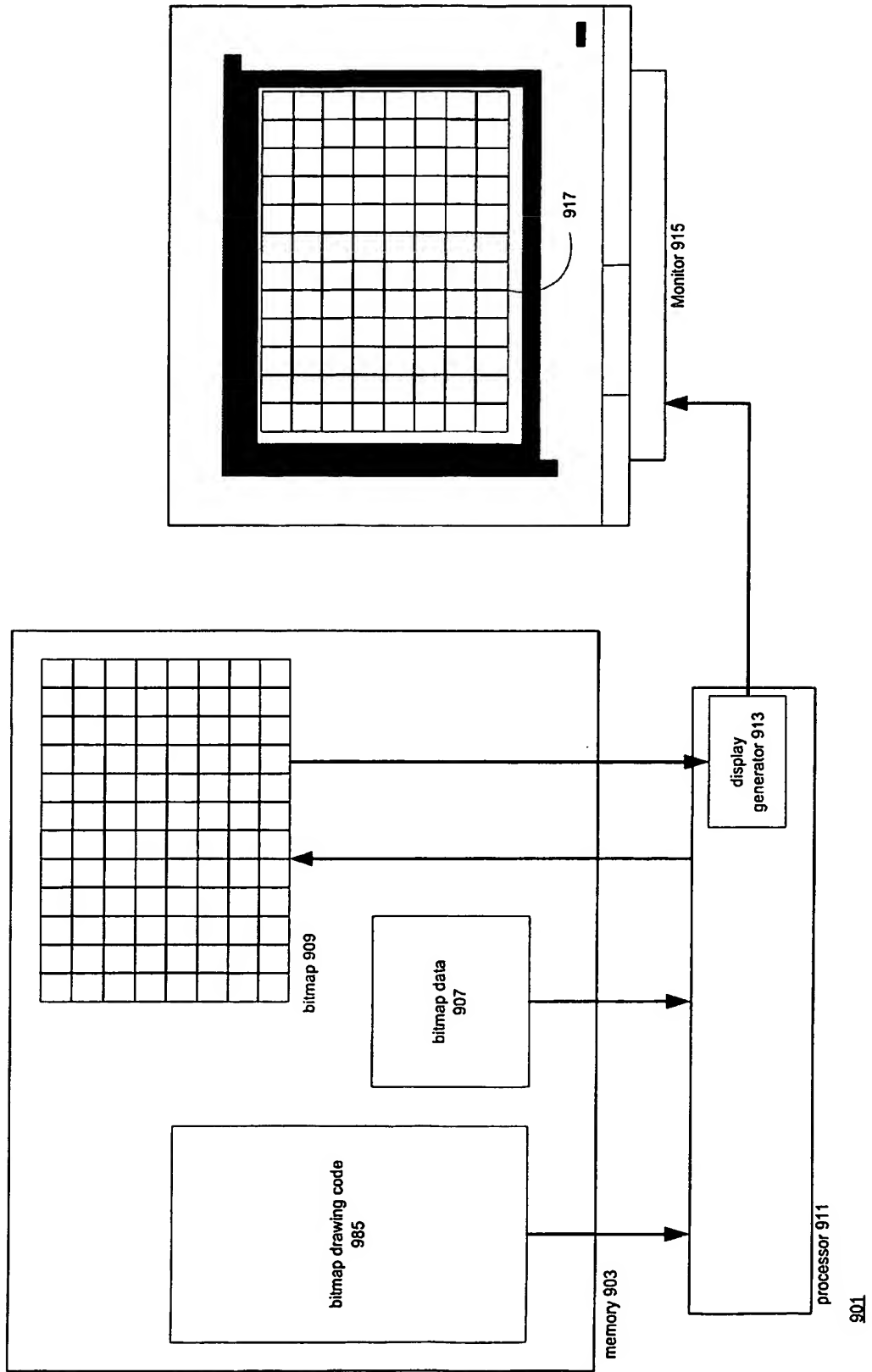


Fig. 9

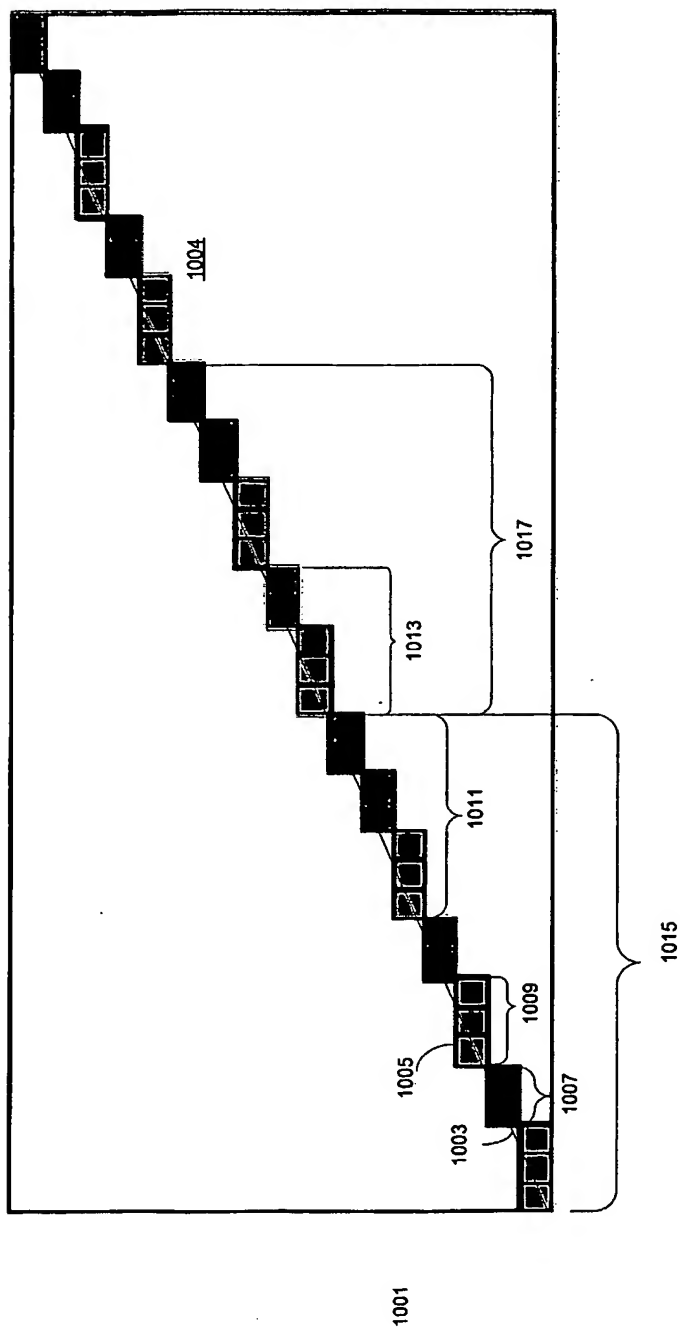


Fig. 10

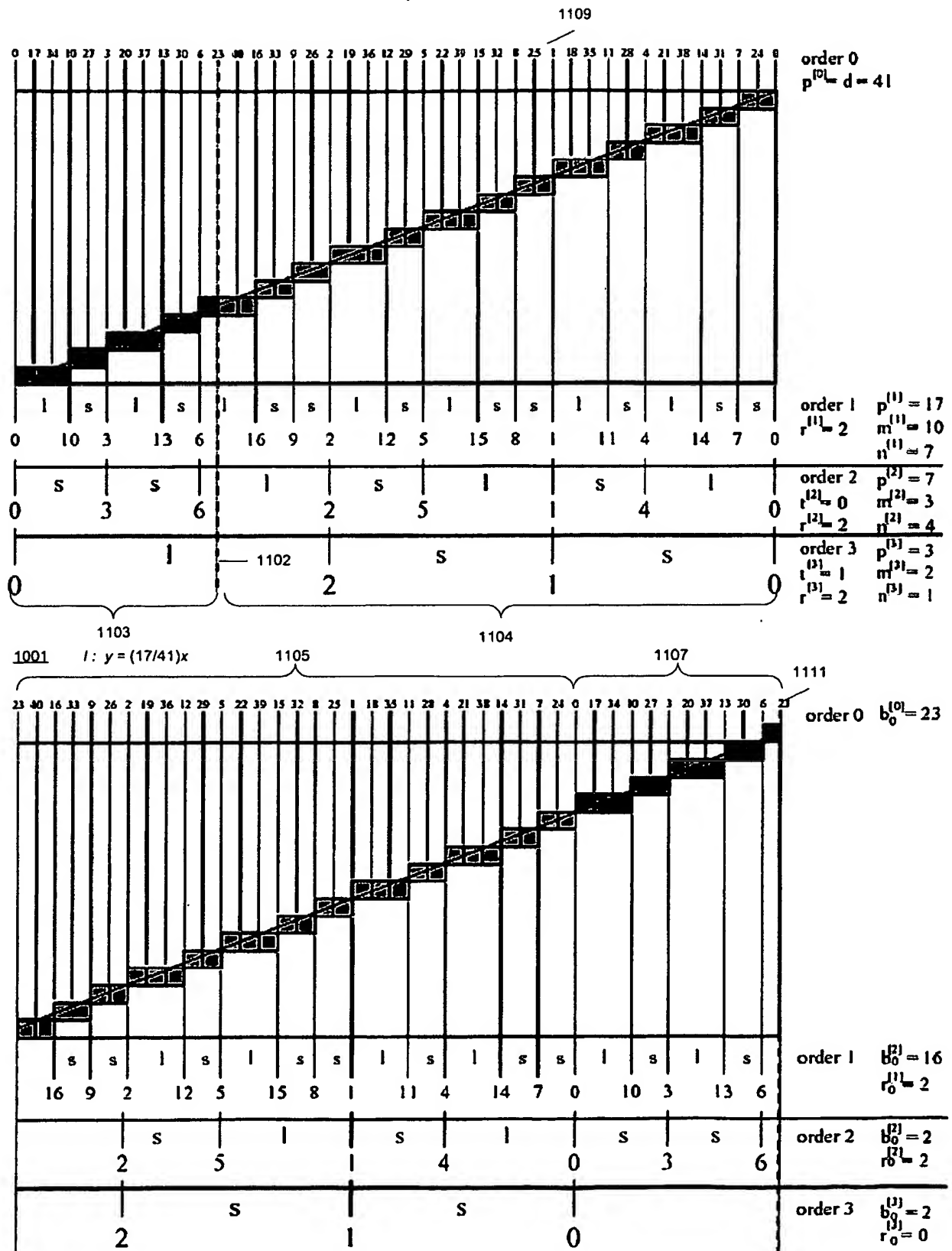


Fig. 11

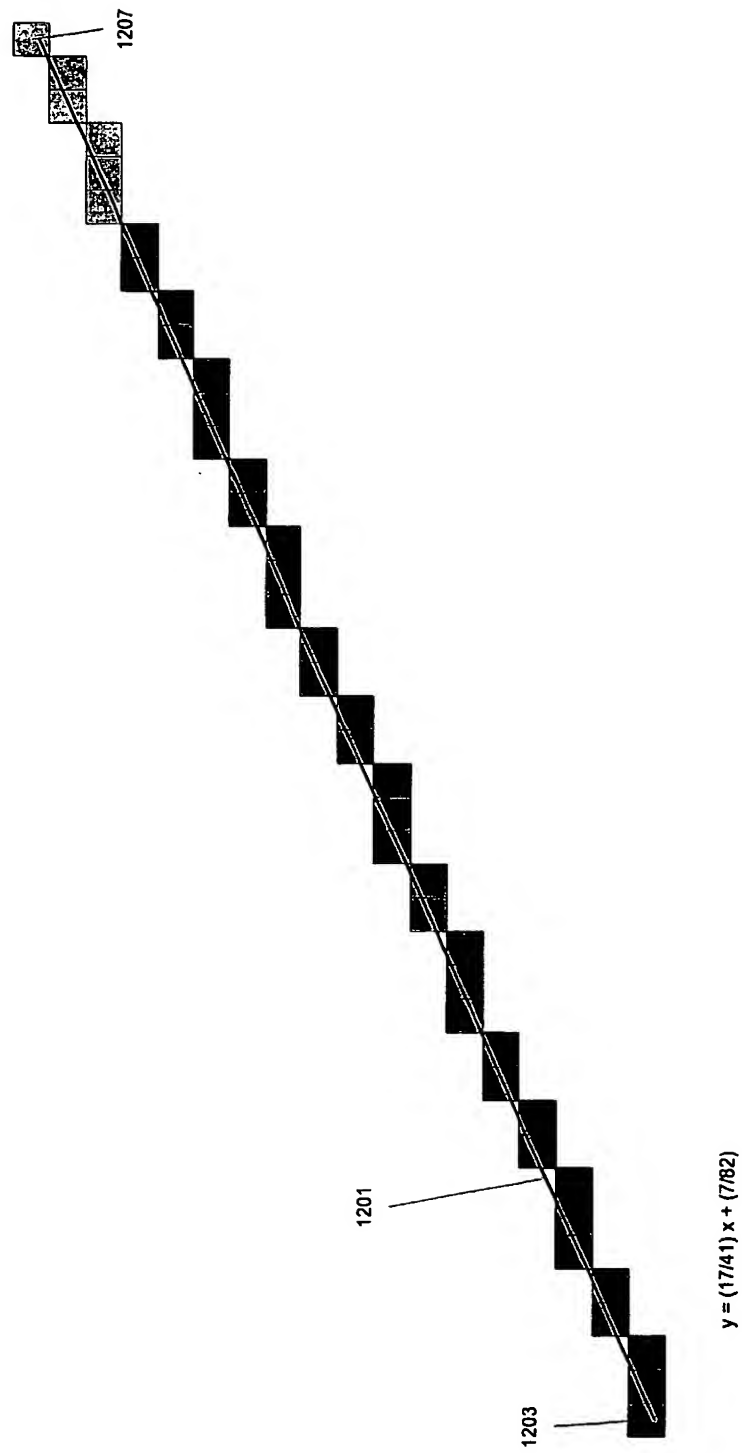


Fig. 12

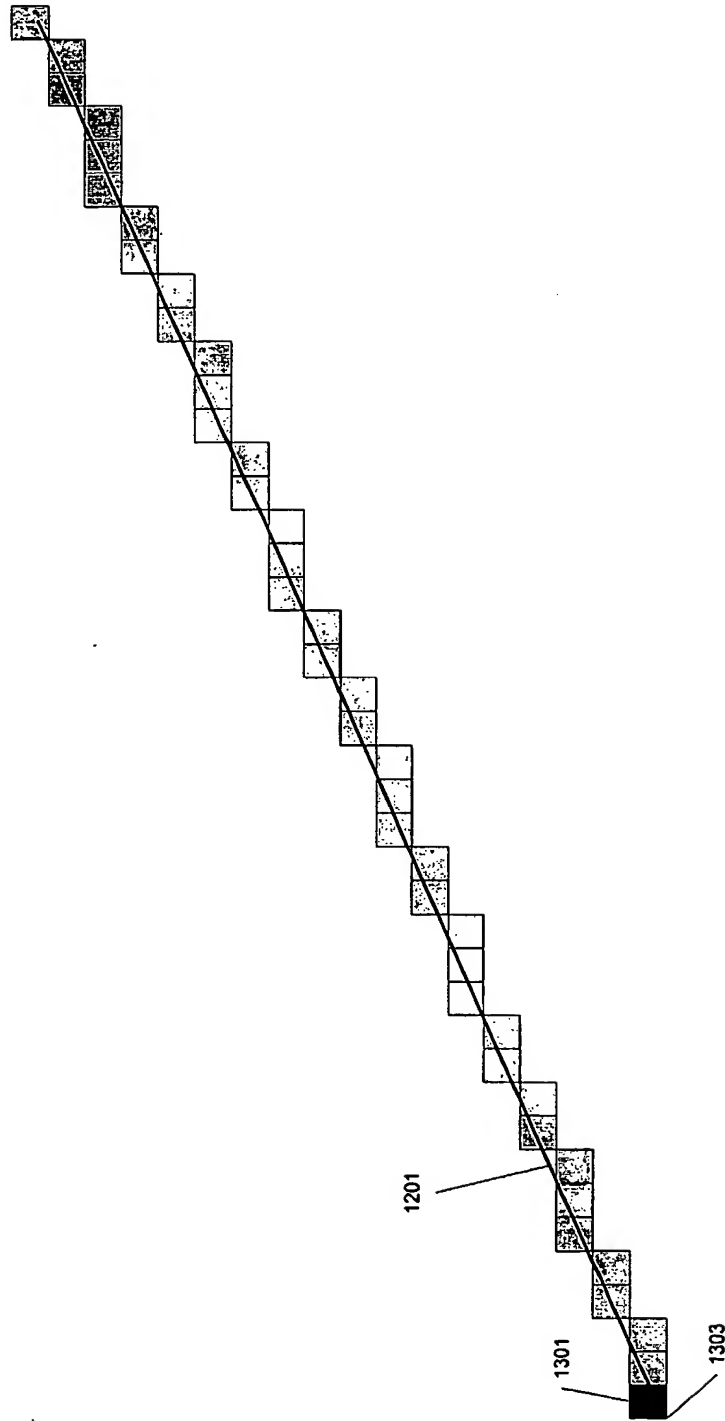


Fig. 13

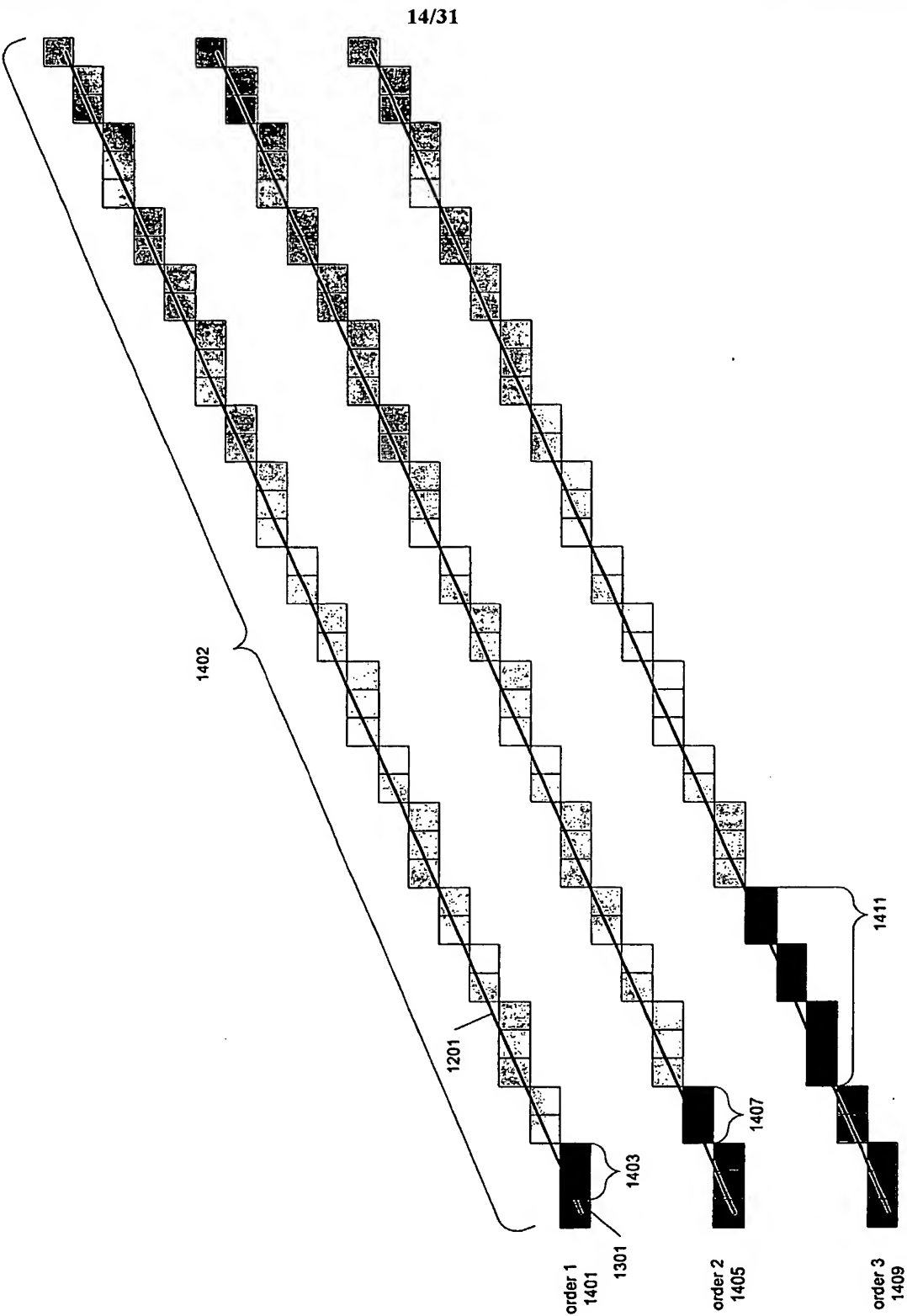
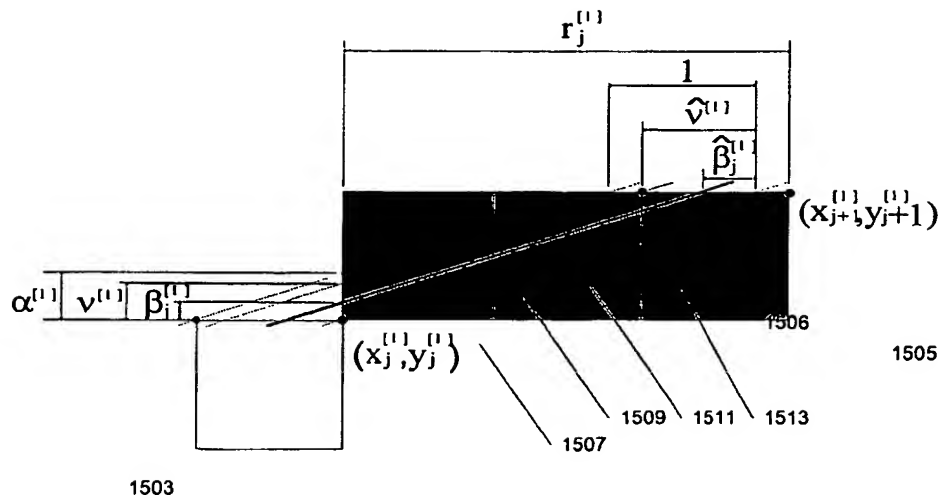
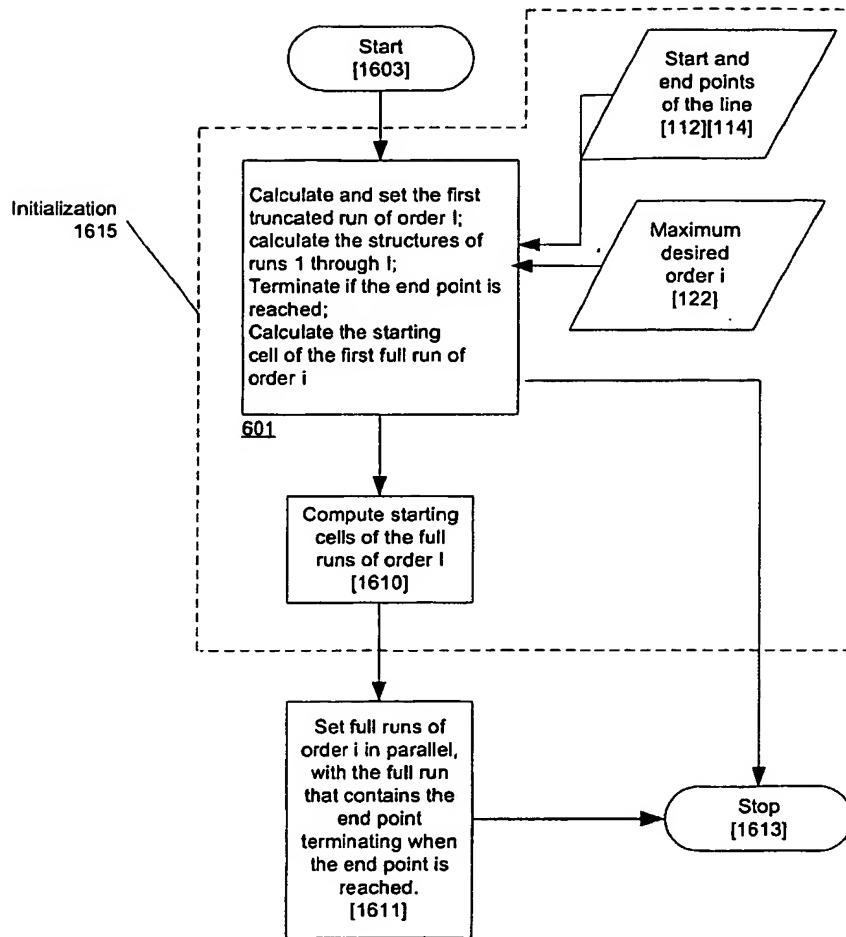


Fig. 14



1501

Fig. 15



1601

Fig. 16

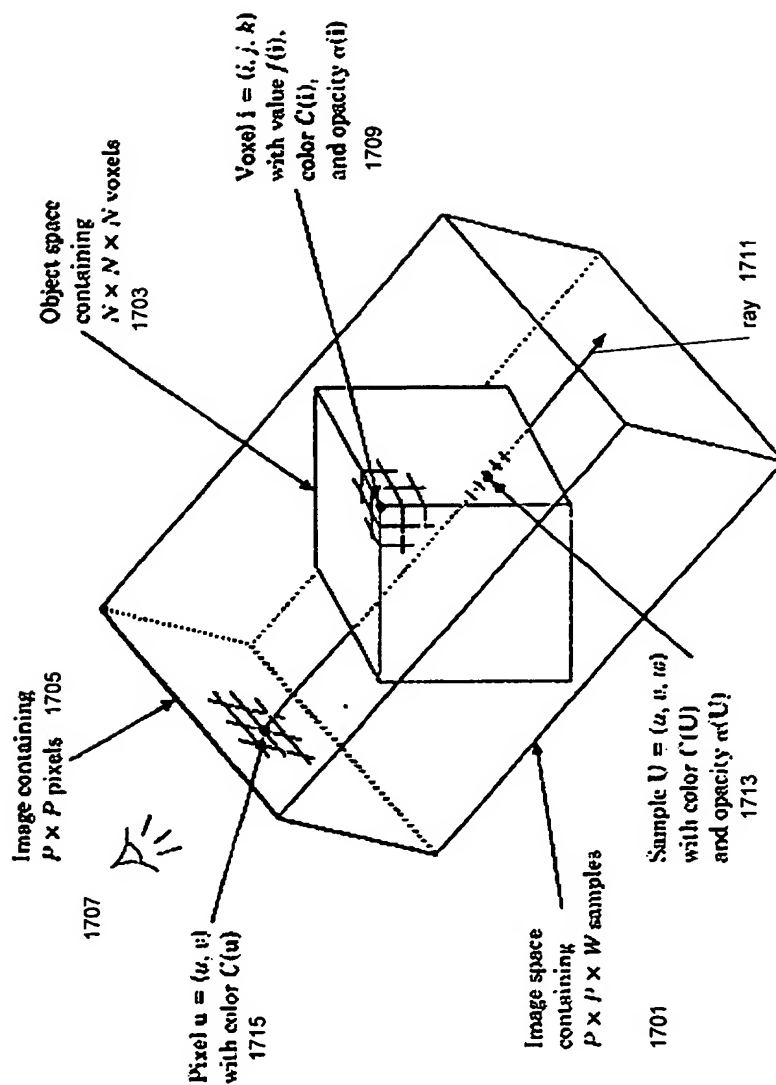


Fig. 17

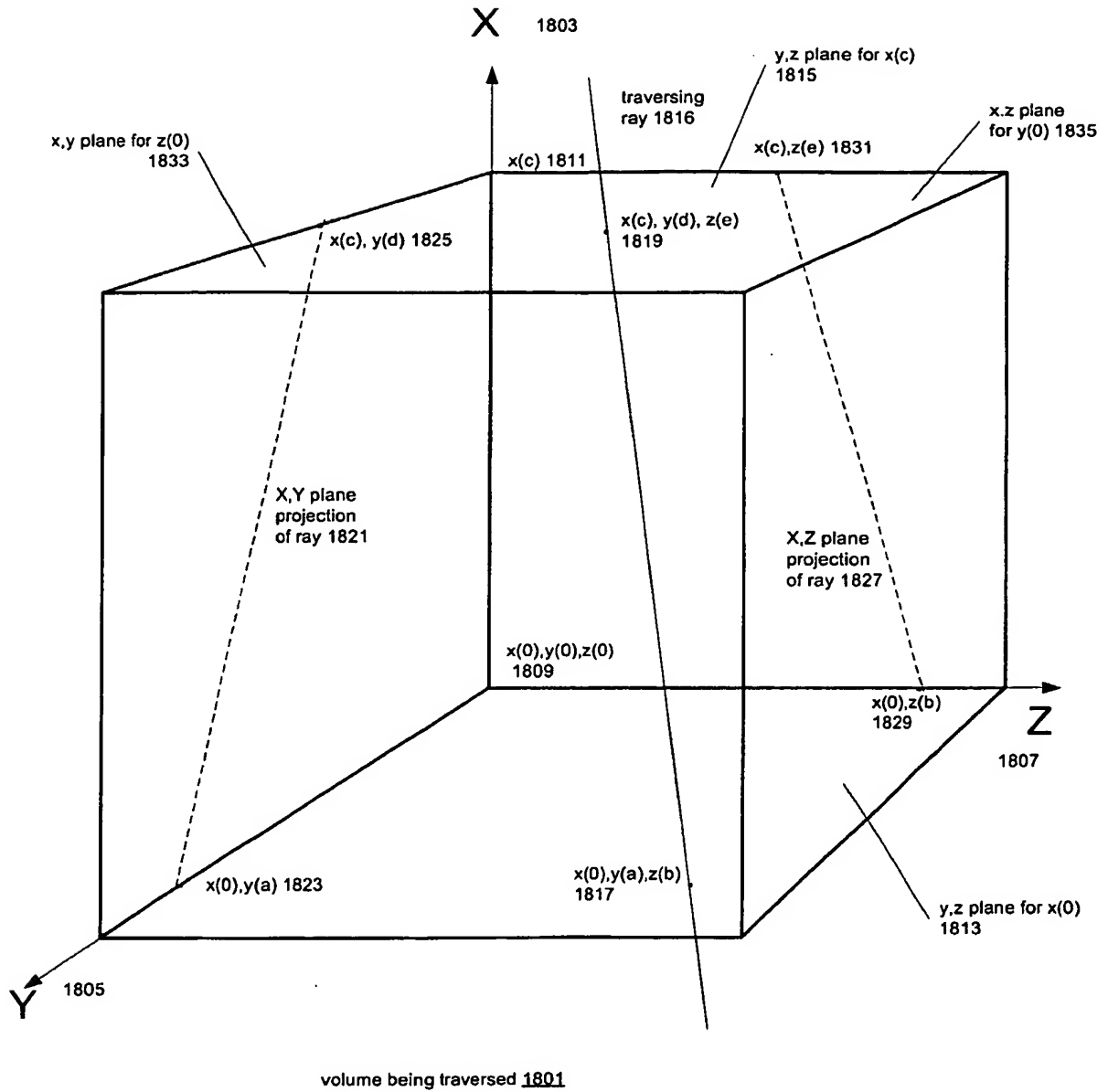
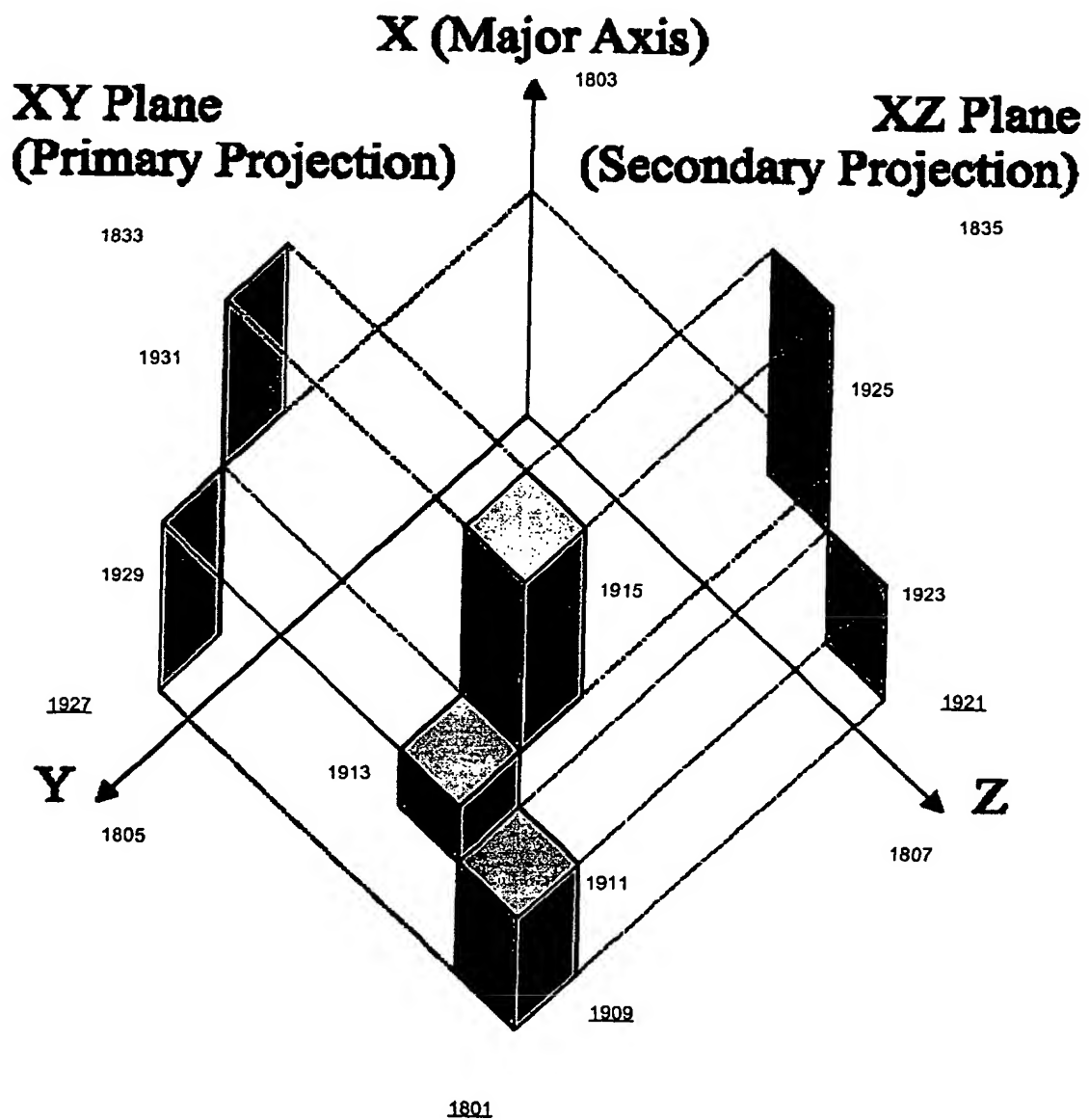


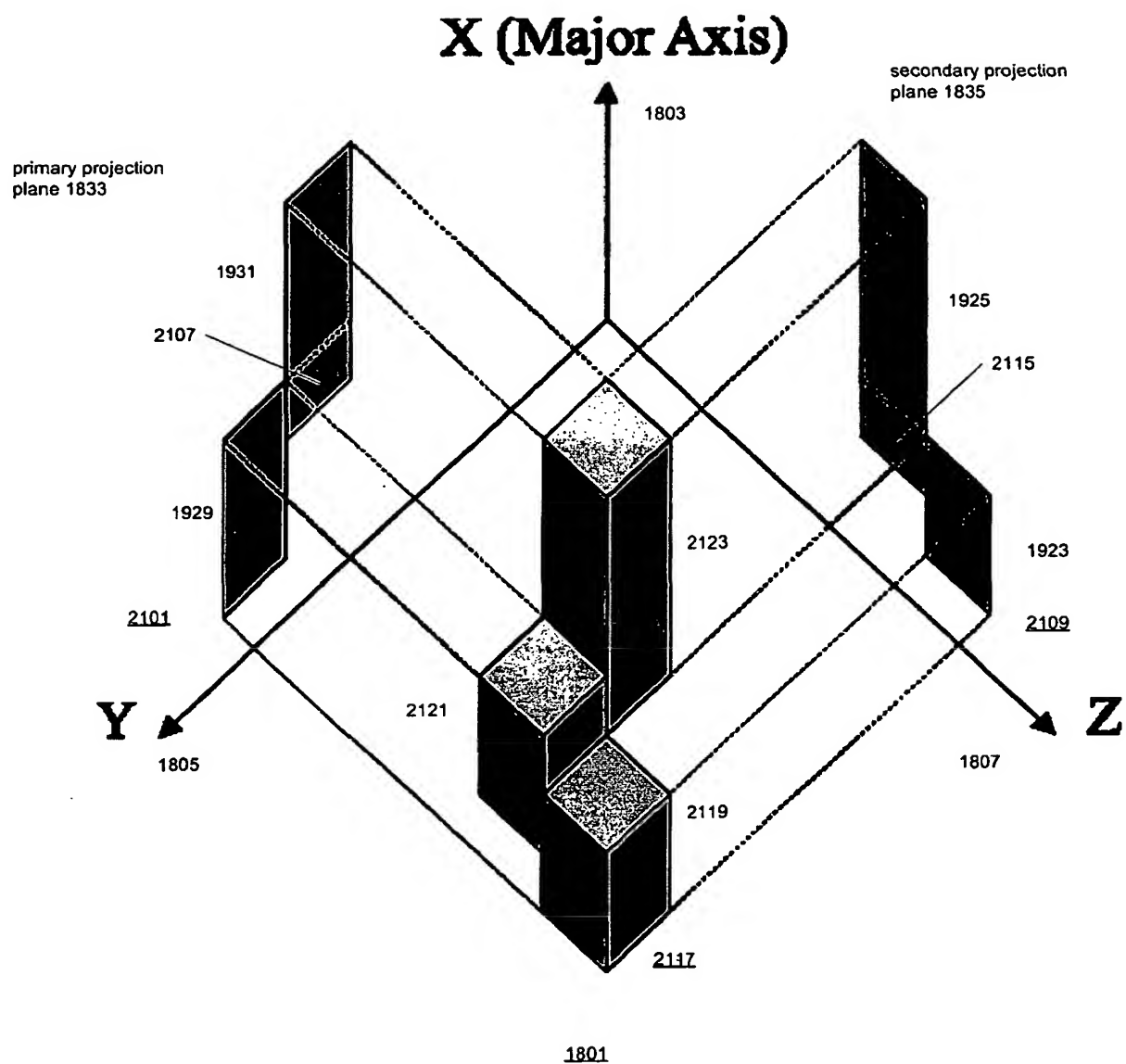
Fig. 18

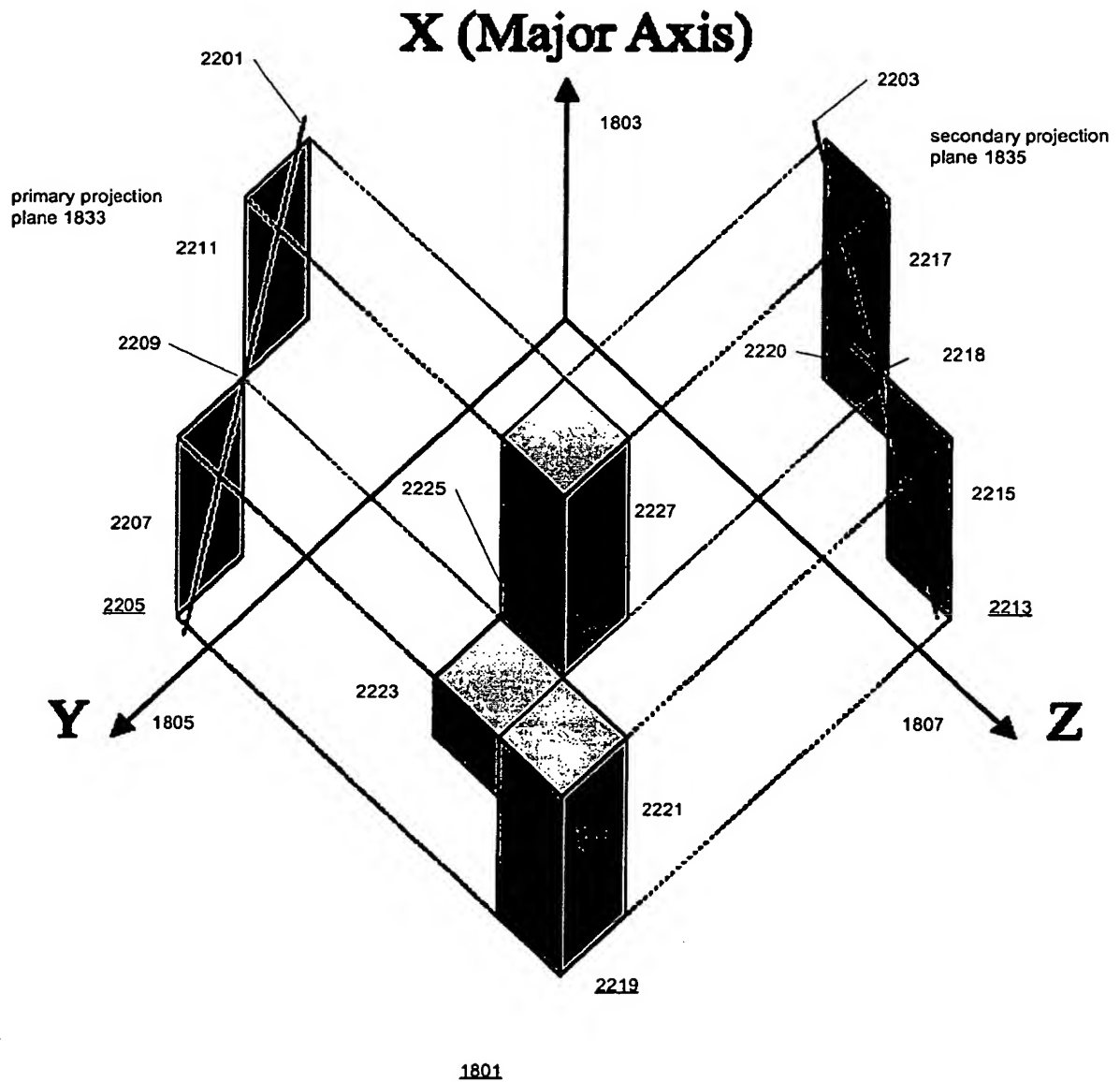
Fig. 19

```

1      (x, y, z) = ray.startPoint
2
3      Get the first run length in each projected ray path.
4      rXY = projectionXY.firstRunLength()
5      rXZ = projectionXZ.firstRunLength()
6
7      while untermiated
8          if rXY < rXZ
9              subdivision.traverseRun(rXY, x, y, z)
10
11             Calculate the position of the next run.
12             x+ = rXY
13             y++
14
15             Shorten the corresponding XZ run.
16             rXZ- = rXY
17
18             Get the next XY run length.
19             rXY = projectionXY.nextRunLength()
20
21         else if rXY > rXZ
22             subdivision.traverseRun(rXZ, x, y, z)
23
24             Calculate the position of the next run.
25             x+ = rXZ
26             z++
27
28             Shorten the corresponding XY run.
29             rXY- = rXZ
30
31             Get the next XZ run length.
32             rXZ = projectionXZ.nextRunLength()
33
34         elseThe XY and XZ runs have the same length.
35             subdivision.traverseRun(rXZ, x, y, z)
36
37             Calculate position of next run.
38             x+ = rXZ
39             y++
40             z++
41
42             Get the next XY and XZ run length.
43             rXY = projectionXY.nextRunLength()
44             rXZ = projectionXZ.nextRunLength()

```

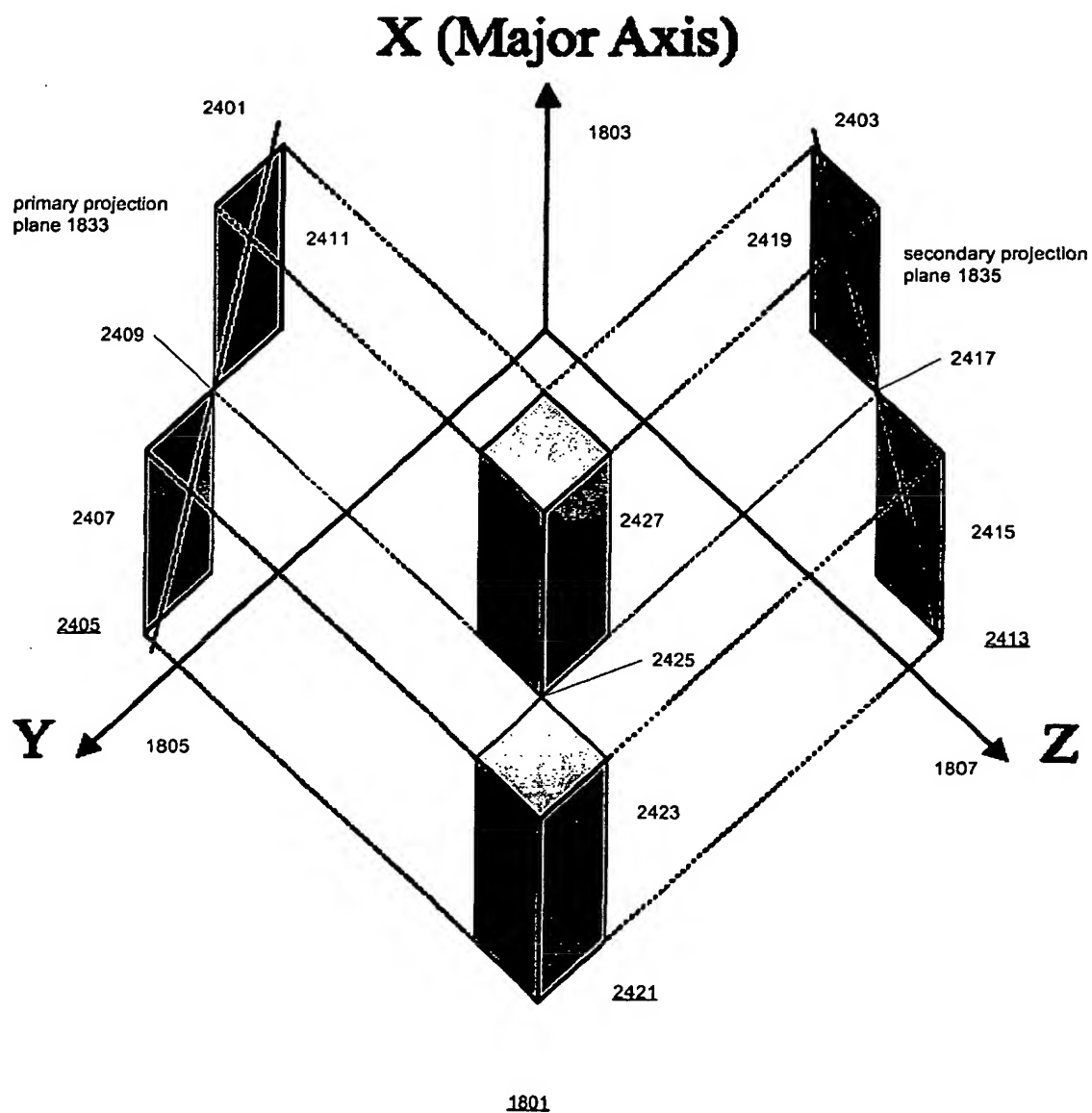
Fig. 21

Fig. 22

```
1  if  $r_{XY} < r_{XZ}$ 
2      if  $projection_{XY}.\beta$  is non-zero
3          No edge intersection.
4           $subdivision.traverseRun$ 
               $(r_{XY} + 1, x - 1, y, z)$ 
5      else
6          Edge intersection.
7           $subdivision.traverseRun(r_{XY}, x, y, z)$ 
8
9       $x+ = r_{XY}$ 
10      $y++$ 
11      $r_{XZ-} = r_{XY}$ 
12      $r_{XY} = projection_{XY}.nextRunLength()$ 
```

2301

Fig. 23

Fig. 24


```
1  if  $r_{XY} == r_{XZ}$ 
2      if  $\text{projection}_{XZ}.\hat{\beta} < \text{projection}_{XY}.\hat{\beta}$ 
3           $\text{subdivision.traverseCell}(x, y, z - 1)$ 
4      else if  $\text{projection}_{XY}.\hat{\beta} < \text{projection}_{XZ}.\hat{\beta}$ 
5           $\text{subdivision.traverseCell}(x, y - 1, z)$ 
6      else  $\text{projection}_{XY}.\hat{\beta} == \text{projection}_{XZ}.\hat{\beta}$ 
7          if  $\text{projection}_{XY}.\hat{\beta}$  is zero
8              No corner intersection.
9               $\text{subdivision.traverseRun}$ 
10                  $(r_{XY} + 1, x - 1, y, z)$ 
11          else
12              Corner intersection.
13               $\text{subdivision.traverseRun}(r_{XY}, x, y, z)$ 
14       $x+ = r_{XY}$ 
15       $r_{XY} = \text{projection}_{XY}.\text{nextRunLength}()$ 
16       $r_{XZ} = \text{projection}_{XZ}.\text{nextRunLength}()$ 
```

2501

Fig. 25

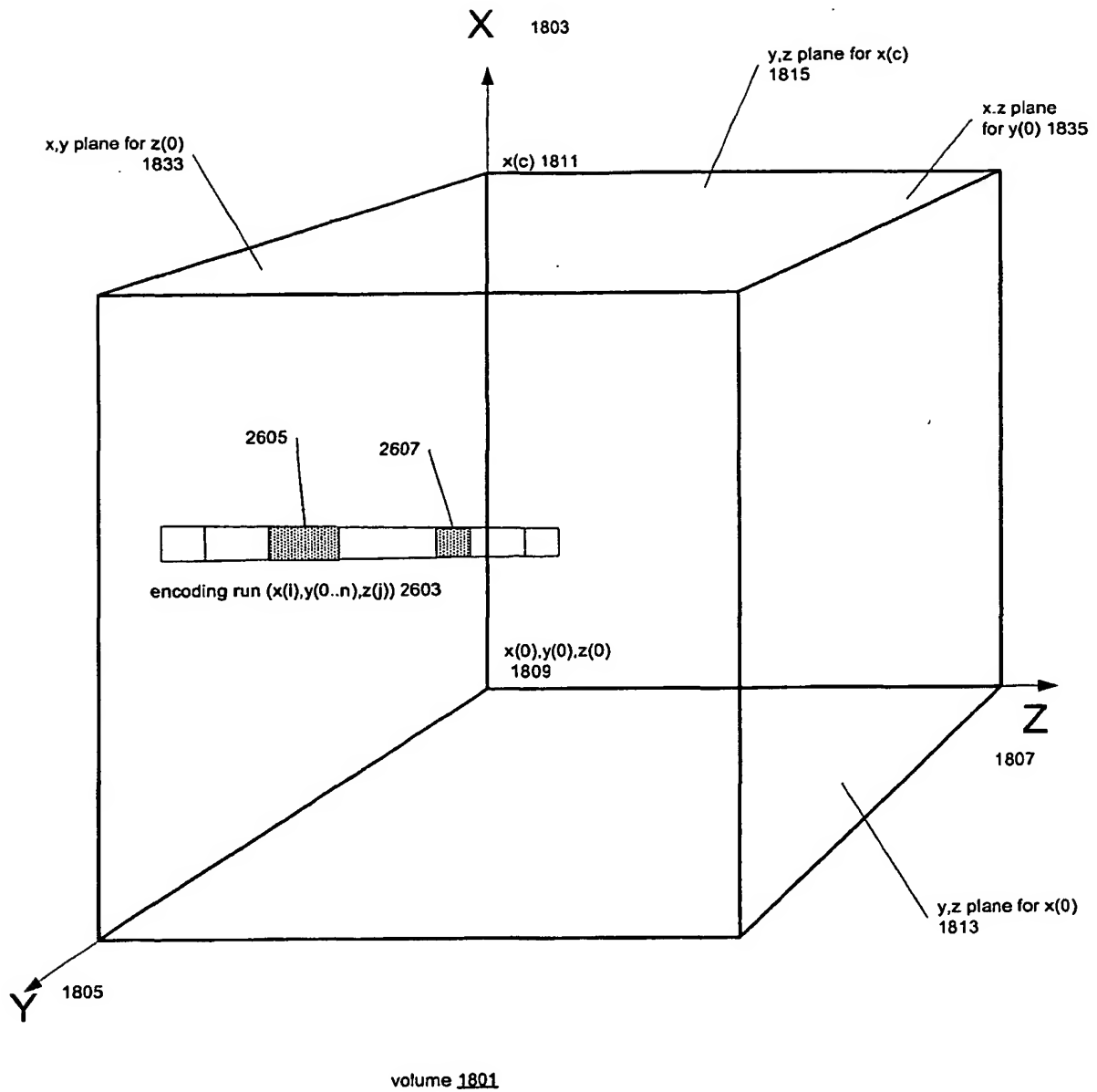


Fig. 26

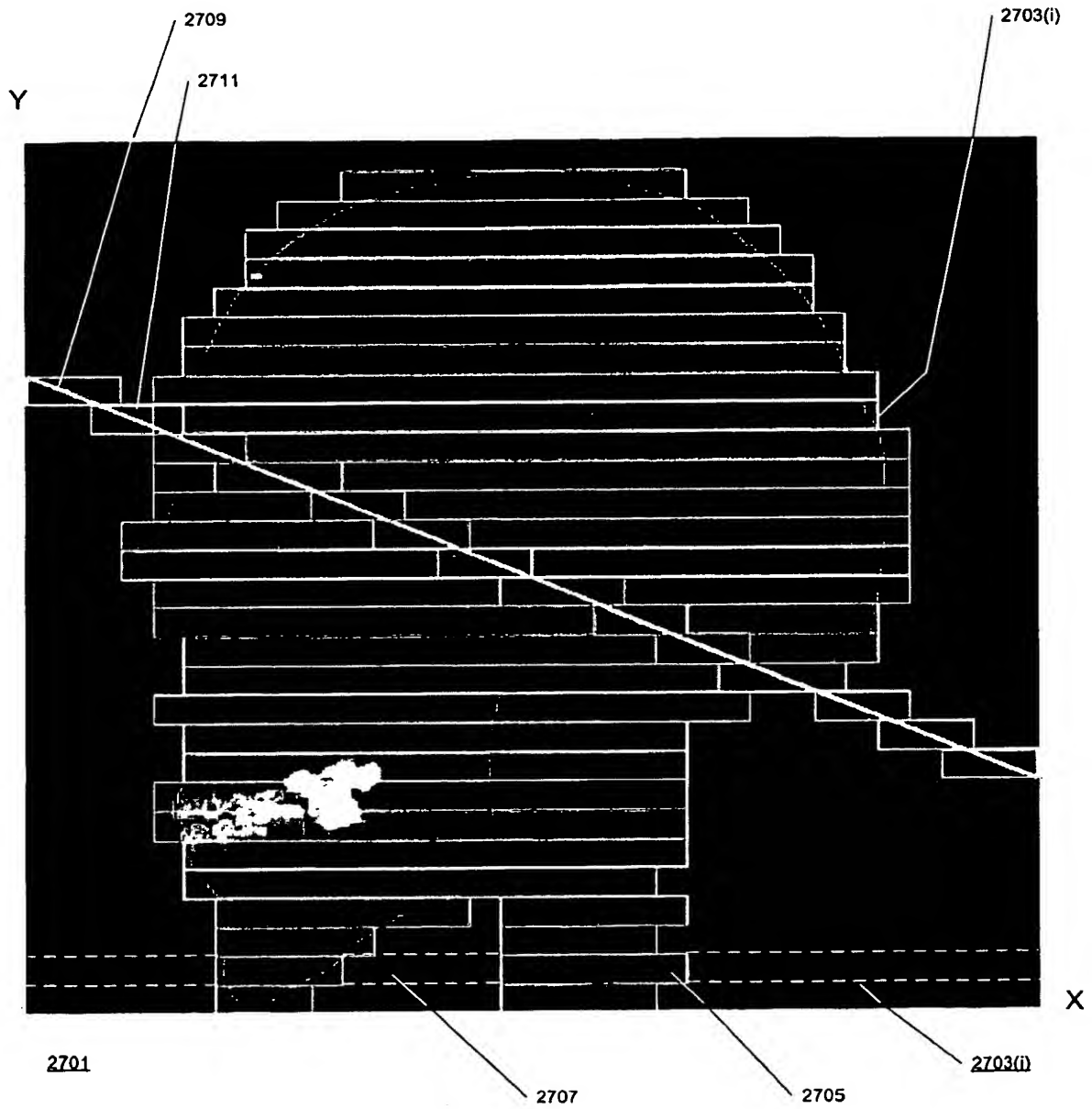


Fig. 27

```
1   For each run in the list
2   for  $i = 0; i < list.length; i++$ 
3       if  $ray.run.end < list.run[i].start$ 
4           No intersection exists
5           return
6
7       if  $ray.run.start < list.run[i].end$ 
8           Intersection exists
9            $x_0 = \max(ray.run.start, list.run[i].start)$ 
10           $x_1 = \min(ray.run.end, list.run[i].end)$ 
11           $subdivision.traverseRun(x_1 - x_0, x_0, y, z)$ 
```

2801

Fig. 28

```

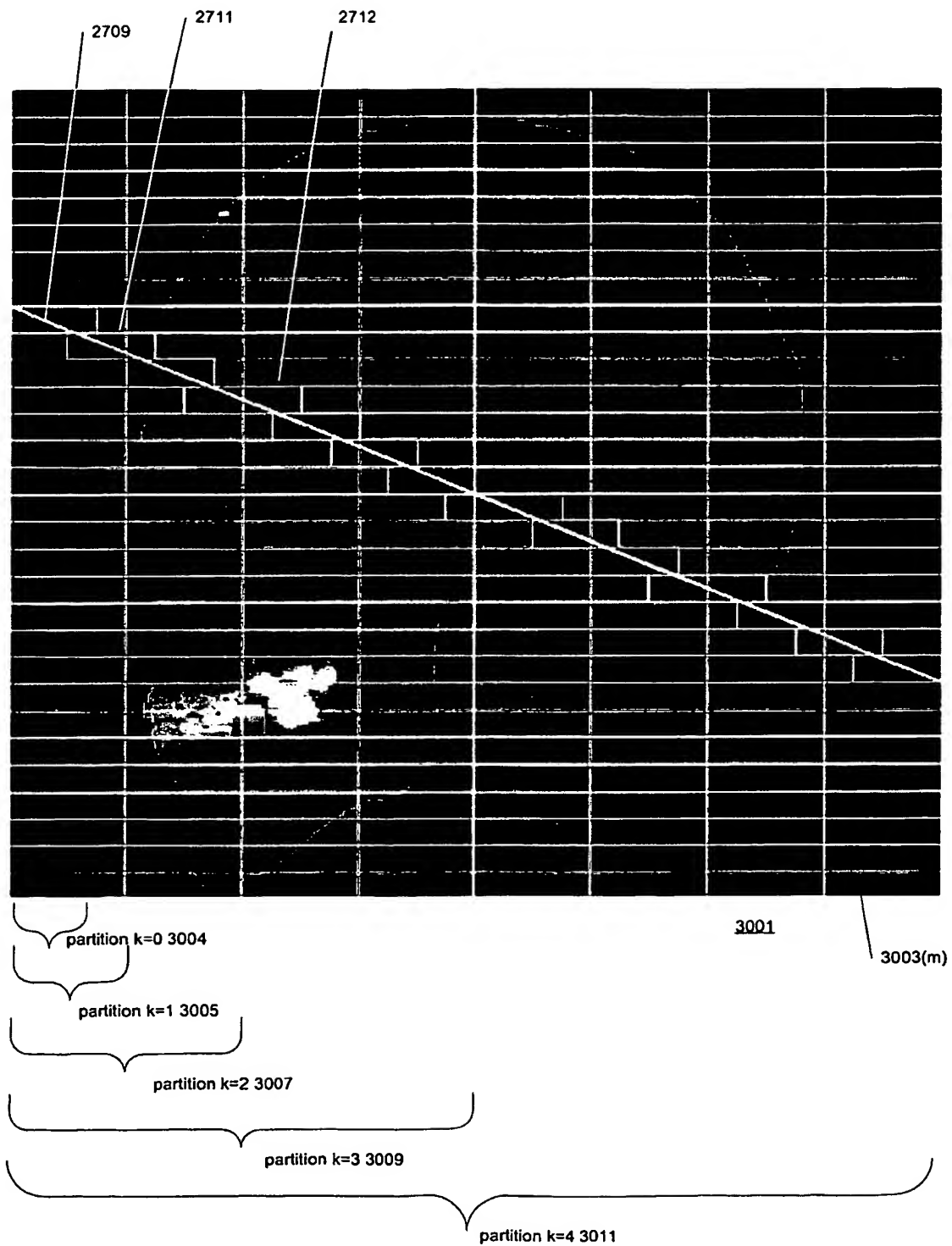
1       $i = 0$ 
2       $j = \text{list.length}$ 
3
4      if  $\text{ray.run.end} < \text{list.run}[i].\text{start}$ 
5          No intersection exists
6          return
7
8      if  $\text{ray.run.start} \geq \text{list.run}[j].\text{end}$ 
9          No intersection exists
10         return
11
12     intersectRunList( $i, j$ )

1 intersectRunList(int  $i$ , int  $j$ )
2     if  $i == j$ 
3         Intersection exists
4          $x_0 = \max(\text{ray.run.start}, \text{list.run}[i].\text{start})$ 
5          $x_1 = \min(\text{ray.run.end}, \text{list.run}[i].\text{end})$ 
6          $\text{subdivision.traverseRun}(x_1 - x_0, x_0, y, z)$ 
7         return
8
9      $j' = \lfloor (i + j) / 2 \rfloor$ 
10     $i' = j' + 1$ 
11
12    if  $\text{ray.run.start} < \text{list.run}[j'].end$ 
13        intersectRunList( $i, j'$ )
14
15    if  $\text{ray.run.end} \geq \text{list.run}[i'].start$ 
16        intersectRunList( $i', j$ )

```

2901

Fig. 29

Fig. 30

```
1   Assume for each run:
2   ray.run.length ≤ partition.size
3   for each run length
4       if ray.run.length > partition.extent
5           Handle tail of run length.
6           if partition is interesting
7               Traverse partition.extent cells.
8               ray.run.length − = partition.extent
9               partition.extent = partition.size
10          Handle head of run length.
11          if partition is interesting
12              Traverse ray.run.length cells.
13          partition.extent − = ray.run.length
```

3101

Fig. 31

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.